Schedae Informaticae Vol. 25 (2016): 103–115
doi: 10.4467/20838476SI.16.008.6189

tfml 2017
theoretical foundations
of machine learning, Kraków

Towards Learning Word Representation

Magdalena Wiercioch
Faculty of Mathematics and Computer Science
Lojasiewicza 6, 30-348 Kraków, Poland
e-mail: *magdalena.wiercioch@ii.uj.edu.pl*

Abstract. Continuous vector representations, as a distributed representations for words have gained a lot of attention in Natural Language Processing (NLP) field. Although they are considered as valuable methods to model both semantic and syntactic features, they still may be improved. For instance, the open issue seems to be to develop different strategies to introduce the knowledge about the morphology of words. It is a core point in case of either dense languages where many rare words appear and texts which have numerous metaphors or similies. In this paper, we extend a recent approach to represent word information. The underlying idea of our technique is to present a word in form of a bag of syllable and letter n-grams. More specifically, we provide a vector representation for each extracted syllable-based and letter-based n-gram, and perform concatenation. Moreover, in contrast to the previous method, we accept n-grams of varied length n. Further various experiments, like tasks-word similarity ranking or sentiment analysis report our method is competitive with respect to other state-of-the-art techniques and takes a step toward more informative word representation construction.

Keywords: representation learning, n-gram model, NLP

1. Introduction

Continuous word representations (embeddings or distributed representations) are found useful for many Natural Language Processing problems such as information

Received: 11 December 2016 / Accepted: 30 December 2016

retrieval or character recognition [1, 2]. Since their quality is strictly connected with aspects of specific language that is being analyzed, each explored issue in this field may also lead to improvement of the particular task where given representation is applied.

Various attempts have been made to investigate learning continuous representations of words in Natural Language Processing [3, 4]. Most of the earliest approaches for learning continuous vectors are based on latent semantic derivations [5, 6]. In particular, its subdomain called distributional semantics where analyzing relationships between a set of documents are considered have been studied extensively in vast majority of papers [7]. In last years neural network researchers have focused on this field [8, 9]. The common drawback of these techniques is the fact they associate a completely distinct vector to each word of the vocabulary. In consequence, the word characteristic information is lost. Take for instance some of dense (highly inflected) languages, i.e. Serbo-Romanian, Romanian which create a challenge for researchers since these languages are seen by linguistics as richly inflected [10]. What is more, although English is not considered as complex, it may be demanding to learn satisfactory representation for corpuses with many rhetorical devices.
On the other hand, the idea of applying more detailed information connected with a given word to a model was presented a few years ago. One of the first approaches to learn representations using fragments of words was character fourgrams-based method introduced by Schütze [4]. In 2003 Bilmes and Kirchhoff investigated factored language models, where a word is viewed as a vector of k factors, such as stems, morphological classes, data-driven clusters [11]. Also, several approaches which rely on a morphological decomposition have been proposed [12, 13]. There is a series of papers which describe models built using recurrent neural networks [14, 15]. Yet another class of methods makes use of convolutional neural networks working on characters. Let us give just a few examples of usage: text classification [16], part-of-speech tagging [17, 18], language modeling [19], sentiment analysis [20] or text normalization [21]. Recently, the concept of using subwords to form a representation appeared [22, 23]. Another work [24] suggests to guide word-embeddings with morphologically annotated data and shows achievement using German in a case study. Also, many papers study syllable-based n-gram methods to model language [25, 26].

In this work, we explore another way to learn word representation using combined character and syllable-level approach. Inspired by the recent works – on continuous bag-of-word model by Mikolov et al. [27] and on using subword information by Bojanowski et al. [28], we show that combining multiple n-grams types enables to capture more word-specific features. This paper is an extension to words vector model proposed by Bojanowski et al. [28]. *Our main goal is to check how extra added syllable information to subword vector representation changes the overall reliability of the model.* What is more, the previous paper uses a very simple scheme where only n-grams of length between 3 and 6 are explored. We does not make such limitations and in consequence our model is able to distinguish short words as well. In order to evaluate our approach, we compare several types of continuous representations, including those made available by other researchers. The evaluation tasks – word similarity ranking analogies and analogy analysis prove the method to be valuable. We achieve improvement for Romanian corpus, too.

The most vital advantage of the proposed model is an attempt to describe word more precisely. For instance, according to our approach "in" has a different position (representation) in word space if it appears as a word itself, and another two locations in case it appears in two independent fragments of another word, e.g. "painting".

2. Model architecture

In this section, we present model to learn specific representation that takes words fragments into account. The proposed representation is an extension of the idea introduced by Bojanowski et al. [28]. Since the model demonstrated by Bojanowski itself is derived from continuous Skip-gram (SG) model introduced by Mikolov et al. in 2013 [8], we first explain how SG works.
Generally, training phase of the Skip-gram model aims at finding word representation that is useful for predicting the surrounding words in a corpus. Let us denote $W = \{w_1, w_2, \ldots, w_S\}$ as the sequence of training words – vocabulary, S – size of vocabulary. The goal of the Skip-gram model is to maximize the average log probability

$$l(W) = \sum_{t=1}^{S} \sum_{c \in C_t} \log p(w_c|w_t),$$

where C_t refers to the context, i.e. the words which surround w_t.
The probability of observing a context word w_c given w_t is parametrized using the word vectors. Given a scoring function s, which maps pairs of (word, context) to value in $\mathbb{R}$, a possible choice to define the probability of a context word is the softmax.

In a basic form the probability of the output context word $Context$ having input $Word$ is defined using the softmax function

$$p(Context|Word) = y_c = \frac{e^{w_c^\top w_t}}{\sum_{j=1}^{S} e^{w_j^\top w_t}},$$

where w_c, w_t, w_j are vector representations of words and y_c is the output of the c-th neuron of the output layer. In practice this formula is not used because of the computational costs. However, an efficient alternative to softmax is Negative sampling, a simple version of Noise Contrastive Estimation (NCE) [29]. While NCE requires from the model to differentiate data from noise by means of logistic regression, Negative Sampling aims only at obtaining high quality representation. Thus, in terms of neural probabilistic language model one may formulate the conditional distribution corresponding to context word c

$$P_\theta^c = \frac{e^{s_\theta(w_t^\top w_c)}}{\sum_{j=1}^{S} e^{s_\theta(w_j^\top w_c)}},$$

where $s_\theta()$ is called scoring function that assesses how the word w_t is compatible with the context w_c.

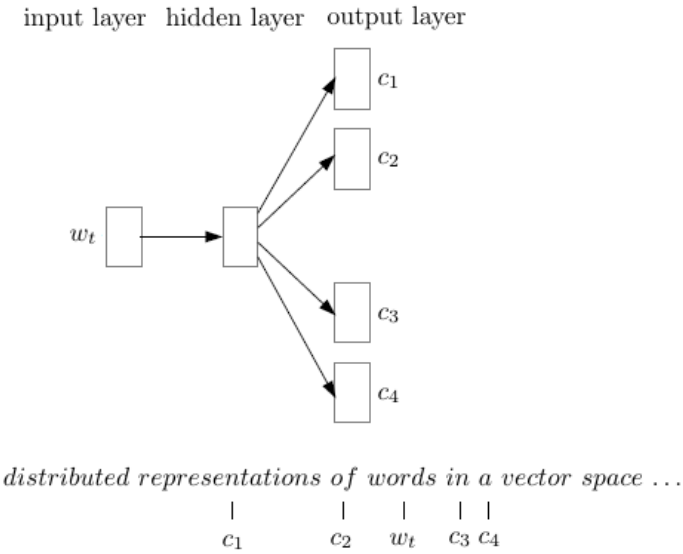


Figure 1. The Skip-gram model architecture with 4 context words considered. The learnt representation enables to predict surrounding words given the current input word "words".

The parametrization for the scoring function is done by taking the scalar product between word and context embeddings: $s(Word, Context) = w_t^\top w_c$[1].

Since the rest of the work uses the concept of letter n-grams and syllable n-grams, they shall be explained.

Letter n-grams

In our paper a letter n-gram is a contiguous sequence of n characters from a given string. As explained in Subsection "Fragmentation model", we use n-grams of several different lengths. In order to distinguish the beginning of a word and the ending of word we append blank spaces to the beginning and the end of the word. Let's take word 'TEST' for example. Note that here the underscore (_) represents the blank space. The following n-grams should be expected.

$$bigrams : \_T, TE, ES, ST, T\_$$

$$trigrams : \_TE, TES, EST, ST\_, T\_\_$$

$$4-grams : \_TES, TEST, EST\_, ST\_\_, T\_\_\_$$

Syllable n-grams

The syllable n-gram is seen as a contiguous sequence of n syllables in a given string. In general, the task of defining the syllable raises some controversy [30, 31]. This work employs the procedure proposed by Daelemans et al. in [32].

[1] Both w_c and w_t are vector representations in $\mathbb{R}^d$.

Fragmentation model

We notice two possible extensions of Bojanowski et al. approach [28]. Firstly, as the authors suggest, the Skip-gram model ignores the internal structure of words. So, they associate a vector representation z_g to each n-gram g. However, we claim it may be insufficient for short and rare words. In this section, we thus propose to extend such a representation by taking syllable n-grams into consideration. Given a word w, let us denote by $G_w = \{1, \ldots, G\}$ the set of letter n-grams which appear in w (as Bojanowski et al. done). Similarly, let $H_w = \{1, \ldots, H\}$ to be the set of syllable n-grams which appear in w. Now, we associate a vector representation z_g to each letter n-gram g and a vector representation z_h to each syllable n-gram h. The new word representation is considered as the direct concatenation of the two vector representations of its n-grams (letter and syllables):

$$z_{new} = [z_g, z_h].$$

In consequence, the scoring function is

$$s(w, c) = \sum_{new \in G_w \cup H_w} z_{new}^{\top} v_c.$$

Secondly, in the model demonstrated by Bojanowski n-grams of length k are only considered, where $3 \leq k \leq 6$. Our analysis show it may negatively affect the final representation reliability. Thus, the upgraded model makes use of n-grams of varied length n.

3. Experiments

Table 1. Spearman's correlation coefficient for the word similarity task.

dataset	RNNLM	NCE	CBoW	Sg	Ft	our
WS353 (en)	0.42	0.45	0.48	0.47	**0.5**	**0.5**
SimVerb-3500 (en)	0.44	0.46	0.44	**0.47**	**0.47**	**0.47**
Sim999 (en)	0.44	0.45	0.45	**0.46**	0.45	0.45
RG65 (en)	0.39	0.4	0.43	0.46	0.46	**0.47**
SGS130 (en)	0.45	0.48	0.5	0.49	**0.5**	**0.5**
YP130 (en)	0.43	0.45	0.44	0.47	**0.48**	**0.48**
Gur30 (ge)	0.45	0.46	0.49	**0.51**	**0.51**	**0.51**
Gur65 (ge)	0.45	0.47	0.52	0.54	0.54	**0.55**
ZG222 (ge)	0.5	0.53	0.53	0.55	**0.56**	**0.56**
RO353 (ro)	0.51	0.55	0.57	0.59	0.59	**0.61**

We conducted a series of experiments to compare the performance of our approach with several strong baseline representations learned on a fixed dataset on different

Table 2. Semantic analogies task results. The accuracy specified as %.

dataset	RNNLM	NCE	CBoW	Sg	Ft	our
WS353 (en)	15.3	24.2	0.23.8	**28**	27.5	27.5
SimVerb-3500 (en)	20.1	26.7	30.6	**34.5**	**34.5**	34
Sim999 (en)	18.3	21.2	**29.8**	24.3	24.8	24.8
RG65 (en)	29.7	35.2	39.1	**42**	**42**	**42**
SGS130 (en)	35.2	41.3	47	**56.1**	**56.1**	**56.1**
YP130 (en)	46.4	42.6	43.6	**56.3**	**56.3**	**56.3**
Gur30 (ge)	37.2	61.2	38.7	**46.7**	**46.7**	**46.7**
Gur65 (ge)	39.8	34.2	44.7	**46.9**	46	46
ZG222 (ge)	41.7	36.2	55.3	**52.6**	**52.6**	52.2
RO353 (ro)	43.9	50	46.6	**60.4**	60.1	60.1

Table 3. Syntactic analogies task results. The accuracy specified as %.

dataset	RNNLM	NCE	CBoW	Sg	Ft	our
WS353 (en)	24.7	30.2	33.5	**40.9**	40.2	40.2
SimVerb-3500 (en)	31.6	33.9	37.2	**52**	**52**	**52**
Sim999 (en)	26	32	**55**	49.8	49.3	49.5
RG65 (en)	35.6	40.2	40.7	**48.9**	**48.9**	**48.9**
SGS130 (en)	38.4	**59**	43.2	49.6	49.6	49.6
YP130 (en)	32.3	37.8	45.8	**50.3**	**50.3**	**50.3**
Gur30 (ge)	30.1	35.2	40.9	**49.3**	**49.3**	**49.3**
Gur65 (ge)	24	35.7	47.3	**62.5**	**62.5**	**62.5**
ZG222 (ge)	38.7	45.3	56.9	**67.2**	**67.2**	67.1
RO353 (ro)	30.6	41.7	**59.2**	53.1	53.1	53.1

tasks. In our experiments we used benchmarks of three languages, i.e. English, German and Romanian. For English, we evaluated word vectors on the following datasets: WS353 [33], SimVerb-3500 [34], Sim999 [35], RG65 [36], SGS130 [37], YP130 [38]. For German, the models were compared on datasets: Gur30, Gur65 [39], ZG222 [40]. For Romanian, the translated version of WS353 was used [41] (RO353). The data contains word pairs along with human-assigned similarity judgements. We compared our approach with 5 baseline representations. These include a model based on recurrent neural network (RNNLM) from 2010 [42] and a method trained using Noise Contrastive Estimation (NCE) presented in [43]. We also took into account two log bilinear methods by Mikolov, i.e. Continuous Bag of Words (CBoW) and mentioned here previously Skip-gram (SG) [8]. Finally, the implementation of the model proposed by Bojanowski et al. (Ft) was examined [28]. Apart from NCE case, we used publicly available codes of mentioned models.

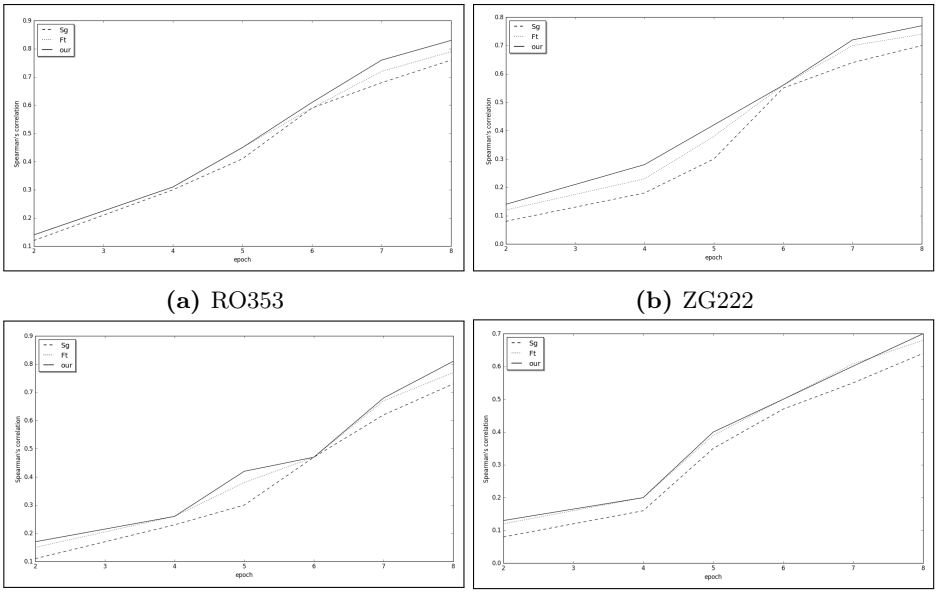
Setup details

In order to provide a reliable comparison, all the methods were trained on the same datasets. For the baseline methods we used default settings presented in papers with

a couple of exceptions. They include a context window of 6 words (both left and right). Additionally, the learning rate was fixed to 3×10^3 and the vector representations had the dimension 200.

Similarity judgement task

The most widely used method of representation quality evaluation is Spearman's rank correlation coefficient [44]. It enables to assess how well the given representations capture word similarity. For instance, "popular" and "famous" are supposed to be closer each other than "trendy" and "fruit". Thus, according to standard techniques, we calculated cosine distance between word pairs in datasets and reported Spearman's rank correlation coefficient between the rankings obtained from the models and human rankings. Table 1 yields the results for the word similarity task. It can be observed that our method slightly outperformed the baseline models in 3 cases. Two of them refer to German and Romanian, so it suggests the proposed technique better describes dense languages (note that German is much more dense than English).



(a) RO353 **(b)** ZG222

(c) SimVerb-3500 **(d)** WS353

Figure 2. The plots of performance versus training epoch for word similarity task.

Word analogy task

Another methods of evaluation are so-called analogy tasks [27]. They enable to assess syntactic and semantic relations between words. In practice, there are sets of questions and each question contains a missing word. The goal is to predict this word. The example of semantic question could be "brother" $\longleftrightarrow$ "sister" ; "grandson" $\longleftrightarrow$ "granddaughter", where the word "granddaughter" has to be predicted. According to [27], it is sufficient to obtain the vector $v = a_{brother} - a_{sister} + a_{grandson}$. We assume the answer is correct if the calculated vector v has high cosine similarity if compared to the good answer. The results for semantic and syntactic analogy tasks are listed

Table 4. Spearman's correlation coefficient for the word similarity task with a much more bigger corpus (200M tokens) and different length of vector representation.

representation length	Sg	Ft	our
200	0.61	**0.65**	**0.65**
300	0.64	0.69	**0.73**
400	0.66	0.71	**0.75**
500	0.7	0.76	**0.81**
600	0.72	0.83	**0.86**

Table 5. Semantic analogies task results with a much more bigger corpus (200M tokens) and different length of vector representation. The accuracy specified as %.

representation length	Sg	Ft	our
200	60.2	65.7	**70.1**
300	63.1	66.4	**68.1**
400	69.7	73.5	**75.2**
500	73.4	77.2	**81.1**
600	77	82.3	**84.8**

in Table 2 and 3, respectively. In fact, our method did not overcome any competing model. Nevertheless, it gave similar results to other Skip-gram based approaches. It shows it may be worth to explore the method's performance on more dense languages.

As one may observe, the previous experiments showed that the most significant results were achieved by Sg, Ft and our approach. Thus, we explored these methods more deeply in the next analysis. First of all, during the experiments we noticed that different models converged at different rates. Figure 2 plots the performance of the word similarity task on selected datasets after a specified number of epochs (2–8). The chart demonstrates that the all three models converge quickly to a satisfactory level of performance. Nevertheless, it appears that our approach yields more reliable results. This suggests that if training was done on more data, the representation could work better. Inspired by this observation, a few other experiments were carried out. We evaluated 3 representations, i.e. Sg, Ft, and our approach. The following tasks were undertaken: word similarity, syntactic and semantic analogies. They were trained on Wikipedia sets which contain 200M tokens[2]. The summary results of evaluations that consider vector space dimensions from 200 to 600 are presented in Table 4, 5 and 6. It is interesting to note that our model is in the vast majority of cases better than Sg and Ft. It performs favorably for either word similarity task (Table 4) or semantic analogies task (Table 5). Although the efficiency of our model

[2] `https://dumps.wikimedia.org/`

Table 6. Syntactic analogies task results with a much more bigger corpus (200M tokens) and different length of vector representation. The accuracy specified as %.

representation length	Sg	Ft	our
200	55.4	57.1	**57.6**
300	58.2	**59.3**	59
400	64.3	**67.8**	69.3
500	69.1	**73.2**	73
600	72.6	76.9	**77.1**

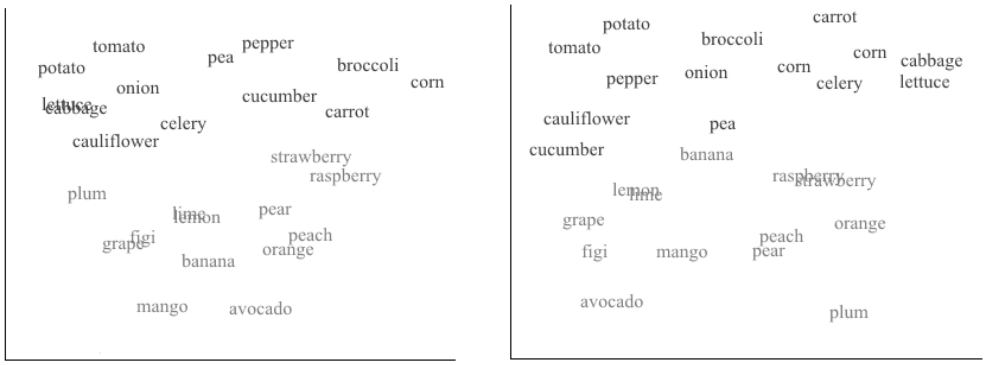


Figure 3. Two dimensional projections of our method and Bojanowski-based (right) word representations. Words associated with "fruit" are colored in grey, words associated with "vegetable" are colored in black. We can see that "fruit" and "vegetable" words are clustered correctly. However, our approach performs slightly better.

on the syntactic analogies task is not strong, it provides some improvements, see Table 6. All in all, the results suggest that our approach benefits from a bigger corpus.

Finally, for our approach and Skip-gram method proposed by Bojanowski et al. we projected the learned word representations into two dimensions using the t-SNE tool [45]. Figure 3 shows projections of the words related to the concept fruit vs. vegetable. The visual inspection demonstrates that all words were assigned to their groups correctly. However, the position of "peach" and "orange" seems to be more adequate if our model is considered.

4. Conclusion

In this paper, we propose method to learn word representations that considers fragments of words, including syllables and characters to built the model. We showed that our method outperforms state-of-the-art approaches on dense languages when tasks such as word similarity ranking or syntactic and semantic analogies are taken into consideration.

Acknowledgment

This research was partially supported by National Centre of Science (Poland) Grants No. 2016/21/N/ST6/01019.

5. References

[1] Miller S., Guinness J., Zamanian A., *Name tagging with word clusters and discriminative training.* In: *Proceedings of HLT*, 2004, pp. 337–342.

[2] Vitz P.C., Winkler B.S., *Predicting the judged similarity of sound of english words.* Journal of Verbal Learning and Verbal Behavior, 1973, 12 (4), pp. 373–388.

[3] Rumelhart D.E., Hinton G.E., Williams R.J., *Neurocomputing: Foundations of research.* MIT Press 1988 pp. 696–699.

[4] Schütze H., *Dimensions of meaning.* In: *Proceedings of the 1992 ACM/IEEE Conference on Supercomputing.* Supercomputing '92, Los Alamitos, CA, USA, IEEE Computer Society Press, 1992, pp. 787–796.

[5] Deerwester S., Dumais S.T., Furnas G.W., Landauer T.K., Harshman R., *Indexing by latent semantic analysis.* Journal of the American Society for Information Science, 1990, 41 (6), pp. 391–407.

[6] Hofmann T., *Probabilistic latent semantic indexing.* In: *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.* SIGIR '99, New York, NY, USA, ACM, 1999, pp. 50–57.

[7] Baroni M., Lenci A., *Distributional memory: A general framework for corpus-based semantics.* December 2010, 36 (4), pp. 673–721.

[8] Mikolov T., Sutskever I., Chen K., Corrado G.S., Dean J., *Distributed representations of words and phrases and their compositionality.* In: Burges C.J.C., Bottou L., Welling M., Ghahramani Z., Weinberger K.Q., eds.: *Advances in Neural Information Processing Systems 26.* Curran Associates, Inc. 2013 pp. 3111–3119.

[9] Pennington J., Socher R., Manning C.D., *Glove: Global vectors for word representation.* In: *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543.

[10] Rehm G., Uszkoreit H., *The Romanian Language in the Digital Age.* Springer Publishing Company, Incorporated, 2012.

[11] Bilmes J.A., Kirchhoff K., *Factored language models and generalized parallel backoff.* In: *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: Companion Volume of the Proceedings of HLT-NAACL 2003–short Papers – Volume 2.* NAACL-Short '03, Stroudsburg, PA, USA, Association for Computational Linguistics, 2003, pp. 4–6.

[12] Botha J.A., Blunsom P., *Compositional Morphology for Word Representations and Language Modelling.* In: *Proceedings of the 31st International Conference on Machine Learning (ICML)*, 2014.

[13] Luong M.T., Socher R., Manning C.D., *Better word representations with recursive neural networks for morphology.* In: *CoNLL*, Sofia, Bulgaria, 2013.

[14] Mikolov T., Sutskever I., Deoras A., Le H.S., Kombrink S., Cernocky J., *Subword language modeling with neural networks.* preprint (http://www. fit. vutbr. cz/imikolov/rnnlm/char. pdf), 2012.

[15] Sutskever I., Martens J., Hinton G.E., *Generating text with recurrent neural networks.* In: *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 2011, pp. 1017–1024.

[16] Zhang X., Zhao J., LeCun Y., *Character-level convolutional networks for text classification.* In: *Advances in Neural Information Processing Systems*, 2015, pp. 649–657.

[17] Ling W., Luís T., Marujo L., Astudillo R.F., Amir S., Dyer C., Black A.W., Trancoso I., *Finding function in form: Compositional character models for open vocabulary word representation.* arXiv preprint arXiv:1508.02096, 2015.

[18] dos Santos C.N., Gatti M., *Deep convolutional neural networks for sentiment analysis of short texts.* In: *COLING*, 2014, pp. 69–78.

[19] Kim Y., Jernite Y., Sontag D., Rush A.M., *Character-aware neural language models.* arXiv preprint arXiv:1508.06615, 2015.

[20] dos Santos C.N., Zadrozny B., *Learning character-level representations for part-of-speech tagging.* In: *ICML*, 2014, pp. 1818–1826.

[21] Chrupała G., *Normalizing tweets with edit scripts and recurrent neural embeddings.* In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Baltimore, Maryland, June 2014, pp. 680–686.

[22] Luong M.T., Manning C.D., *Achieving open vocabulary neural machine translation with hybrid word-character models.* arXiv preprint arXiv:1604.00788, 2016.

[23] Sennrich R., Haddow B., Birch A., *Neural machine translation of rare words with subword units.* In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7–12, 2016, Berlin, Germany, Volume 1: Long Papers*, 2016.

[24] Cotterell R., Schütze H., *Morphological word-embeddings.* In: *Proc. of NAACL*, 2015.

[25] Sakamoto N., Yamamoto K., Nakagawa S., *Combination of syllable based n-gram search and word search for spoken term detection through spoken queries and iv/oov classification.* Dec 2015, pp. 200–206.

[26] Wechsler M., Munteanu E., Schäuble P., *New techniques for open-vocabulary spoken document retrieval.* In: *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.* SIGIR '98, New York, NY, USA, ACM, 1998, pp. 20–27.

[27] Mikolov T., Chen K., Corrado G., Dean J., *Efficient estimation of word representations in vector space.* arXiv preprint arXiv:1301.3781, 2013.

[28] Bojanowski P., Grave E., Joulin A., Mikolov T., *Enriching word vectors with subword information.* arXiv preprint arXiv:1607.04606, 2016.

[29] Gutmann M.U., Hyvärinen A., *Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics.* Journal of Machine Learning Research, 2012, 13 (Feb), pp. 307–361.

[30] Crystal D., *Dictionary of linguistics and phonetics.* vol. 30. John Wiley & Sons, 2011.

[31] Mayer T., *Toward a totally unsupervised, language-independent method for the syllabification of written texts.* In: *Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology*, Association for Computational Linguistics, 2010, pp. 63–71.

[32] Daelemans W., van den Bosch A., Generalization performance of backpropagation learning on a syllabification task. In: *Proceedings of the 3rd Twente Workshop on Language Technology*, Universiteit Twente, Enschede, 1992, pp. 27–38.

[33] Finkelstein L., Gabrilovich E., Matias Y., Rivlin E., Solan Z., Wolfman G., Ruppin E., *Placing search in context: The concept revisited.* In: *Proceedings of the 10th international conference on World Wide Web*, ACM, 2001, pp. 406–414.

[34] Gerz D., Vulić I., Hill F., Reichart R., Korhonen A., *SimVerb-3500: A Large-Scale Evaluation Set of Verb Similarity.* In: *EMNLP*, 2016.

[35] Hill F., Reichart R., Korhonen A., *Simlex-999: Evaluating semantic models with (genuine) similarity estimation.* Computational Linguistics, 2016.

[36] Rubenstein H., Goodenough J.B., *Contextual correlates of synonymy.* October 1965, 8(10), pp. 627–633.

[37] Szumlanski S.R., Gomez, F., Sims V.K., *A new set of norms for semantic relatedness measures.* In: *ACL (2)*, 2013, pp. 890–895.

[38] Yang D., Powers D.M., *Verb similarity on the taxonomy of WordNet.* Masaryk University, 2006.

[39] Gurevych I., *Using the structure of a conceptual network in computing semantic relatedness.* In: *International Conference on Natural Language Processing*, Springer, 2005, pp. 767–778.

[40] Zesch T., Gurevych I., *Automatically creating datasets for measures of semantic relatedness.* In: *Proceedings of the Workshop on Linguistic Distances.* LD '06, Stroudsburg, PA, USA, Association for Computational Linguistics, 2006, pp. 16–24.

[41] Hassan S., Mihalcea R., *Cross-lingual semantic relatedness using encyclopedic knowledge.* In: *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*, Association for Computational Linguistics, 2009, pp. 1192–1201.

[42] Mikolov T., Karafiát M., Burget L., Cernockỳ J., Khudanpur S., *Recurrent neural network based language model.* In: *Interspeech.* vol. 2., 2010, pp. 3.

[43] Mnih A., Teh Y.W., *A fast and simple algorithm for training neural probabilistic language models.* arXiv preprint arXiv:1206.6426, 2012.

[44] Spearman C., *The proof and measurement of association between two things.* American Journal of Psychology, 1904, 15, pp. 88–103.

[45] Maaten L.v.d., Hinton G., *Visualizing data using t-sne.* Journal of Machine Learning Research, 2008, 9 (Nov), pp. 2579–2605.